Please type a plus sign (+) inside this box → +

04-03-00

A

PTO/SB/05 (4/98)
Approved for use through 09/30/2000 OMB 0651-0032
Patent and Trademark Office U.S DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

jc515 U.S. PTO 03/31/00

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1 53(b))

Attorney Docket No.	042390.P8107
First Inventor or Application Identifier	Carl M. Ellison
Title	GENERATING ISOLATED BUS CYCLES FOR ISOLATED EXECUTION
Express Mail Label No.	EL466333089US

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents

ADDRESS TO: Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231

jc511 U.S. PTO 09/538954 03/31/00

1. ☒ Fee Transmittal Form
(Submit an original, and a duplicate for fee processing)

2. ☒ Specification [Total Pages 34]
(preferred arrangement set forth below)
- Descriptive title of the Invention
- Cross References to Related Applications
- Statement Regarding Fed sponsored R & D
- Reference to Microfiche Appendix
- Background of the Invention
- Brief Summary of the Invention
- Brief Description of the Drawings (if filed)
- Detailed Description
- Claim(s)
- Abstract of the Disclosure

3. ☒ Drawing(s) (35 U.S.C. 113) [Total Sheets 8]

4. Oath or Declaration [Total Pages 6]
a. ☐ Newly executed (original copy)
b. ☐ Copy from a prior application (37 C.F.R. § 1.63(d)) (for continuation/divisional with Box 16 completed)
i. ☐ DELETION OF INVENTOR(S)
Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR §§ 1.63(d)(2) and 1.33(b).

5. ☐ Microfiche Computer Program (Appendix)

6. Nucleotide and/or Amino Acid Sequence Submission (if applicable, all necessary)
a. ☐ Computer Readable Copy
b. ☐ Paper Copy (identical to computer copy)
c. ☐ Statement verifying identity of above copies

*NOTE FOR ITEMS 1 & 13: IN ORDER TO BE ENTITLED TO PAY SMALL ENTITY FEES, A SMALL ENTITY STATEMENT IS REQUIRED (37 C.F.R. § 1.27), EXCEPT IF ONE FILED IN A PRIOR APPLICATION IS RELIED UPON (37 C.F.R. § 1.28).

ACCOMPANYING APPLICATION PARTS

7. ☐ Assignment Papers (cover sheet & document(s))
8. ☐ 37 C.F.R. § 3.73(b) Statement (when there is an assignee) ☐ Power of Attorney
9. ☐ English Translation Document (if applicable)
10. ☐ Information Disclosure Statement (IDS)/PTO - 1449 ☐ Copies of IDS Citations
11. ☐ Preliminary Amendment
12. ☒ Return Receipt Postcard (MPEP 503) (Should be specifically itemized)
13. ☐ *Small Entity Statement(s) ☐ Statement filed in prior application, Status still proper and desired
14. ☐ Certified Copy of Priority Document(s) (if foreign priority is claimed)
15. ☐ Other: .....

16. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment:
☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No: _____
Prior application Information: Examiner _____ Group/Art Unit: _____
For CONTINUATION or DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 4b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts

17. CORRESPONDENCE ADDRESS

☐ Customer Number or Bar Code Label (Insert Customer No. or Attach bar code label here) or ☒ Correspondence address below

Name	BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP				
Address	12400 Wilshire Boulevard, Seventh Floor				
City	Los Angeles	State	California	Zip Code	90025
Country	U.S.A.	Telephone	(714) 557-3800	Fax	(714) 557-3347

Name (Print/Type)	Thinh V. Nguyen, Reg. No. 42,034
Signature	Date 03/31/00

Burden Hour Statement: This form is estimated to take 0.2 hours to complete Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS SEND TO Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

Our Ref. No. 042390.P8107
Express Mail No.: EL466333089US

UNITED STATES PATENT APPLICATION

FOR

GENERATING ISOLATED BUS CYCLES FOR ISOLATED EXECUTION

INVENTORS:

Carl M. Ellison
Roger A. Golliver
Howard C. Herbert
Derrick C. Lin
Francis X. McKeen
Gil Neiger
Ken Reneris
James A. Sutton
Shreekant S. Thakkar
Millind Mittal

PREPARED BY:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 Wilshire Blvd., 7th Floor
Los Angeles, CA 90025-1026
(714) 557-3800

BACKGROUND

1. Field of the Invention

This invention relates to microprocessors. In particular, the invention relates to processor security.

2. Description of Related Art

Advances in microprocessor and communication technologies have opened up many opportunities for applications that go beyond the traditional ways of doing business. Electronic commerce (E-commerce) and business-to-business (B2B) transactions are now becoming popular, reaching the global markets at a fast rate. Unfortunately, while modern microprocessor systems provide users convenient and efficient methods of doing business, communicating and transacting, they are also vulnerable to unscrupulous attacks. Examples of these attacks include virus, intrusion, security breach, and tampering, to name a few. Computer security, therefore, is becoming more and more important to protect the integrity of the computer systems and increase the trust of users.

Threats caused by unscrupulous attacks may be in a number of forms. Attacks may be remote without requiring physical accesses. An invasive remote-launched attack by hackers may disrupt the normal operation of a system connected to thousands or even millions of users. A virus program may corrupt code and/or data of a single-user platform.

Existing techniques to protect against attacks have a number of drawbacks. Anti-virus programs can only scan and detect known viruses. Most anti-virus programs use a weak policy in which a file or program is assumed good until proved bad. For many

security applications, this weak policy may not be appropriate. In addition, most anti-virus programs are used locally where they are resident in the platform. This may not be suitable in a group work environment. Security co-processors or smart cards using cryptographic or other security techniques have limitations in speed performance, memory capacity, and flexibility. Redesigning operating systems creates software compatibility issues and causes tremendous investment in development efforts.

BRIEF DESCRIPTION OF THE DRAWINGS

The features and advantages of the present invention will become apparent from the following detailed description of the present invention in which:

Figure 1A is a diagram illustrating a logical operating architecture according to one embodiment of the invention.

Figure 1B is a diagram illustrating accessibility of various elements in the operating system and the processor according to one embodiment of the invention.

Figure 1C is a diagram illustrating a computer system in which one embodiment of the invention can be practiced.

Figure 2A is a diagram illustrating the isolated execution circuit shown in Figure 1C according to one embodiment of the invention.

Figure 2B is a diagram illustrating the isolated bus cycle generator shown in Figure 2A according to one embodiment of the invention.

Figure 3 is a diagram illustrating an access generator circuit according to one embodiment of the invention.

Figure 4 is a diagram illustrating a bus cycle decoder according to one embodiment of the invention.

Figure 5 is a flowchart illustrating a process to generate isolated bus cycles for isolated execution according to one embodiment of the invention.

DETAILED DESCRIPTION

In the following description, for purposes of explanation, numerous details are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that these specific details are not required in order to practice the present invention. In other instances, well-known electrical structures and circuits are shown in block diagram form in order not to obscure the present invention.

ARCHITECTURE OVERVIEW

One principle for providing security in a computer system or platform is the concept of an isolated execution architecture. The isolated execution architecture includes logical and physical definitions of hardware and software components that interact directly or indirectly with an operating system of the computer system or platform. An operating system and the processor may have several levels of hierarchy, referred to as rings, corresponding to various operational modes. A ring is a logical division of hardware and software components that are designed to perform dedicated tasks within the operating system. The division is typically based on the degree or level of security and/or protection. For example, a ring-0 is the innermost ring, being at the highest level of the hierarchy. Ring-0 encompasses the most critical, security-sensitive components. In addition, modules in Ring-0 can also access to lesser privileged data, but not vice versa. Ring-3 is the outermost ring, being at the lowest level of the hierarchy. Ring-3 typically encompasses users or applications level and has the least security protection. Ring-1 and ring-2 represent the intermediate rings with decreasing levels of security and/or protection.

Figure 1A is a diagram illustrating a logical operating architecture 50 according to one embodiment of the invention. The logical operating architecture 50 is an abstraction

of the components of an operating system and the processor. The logical operating architecture 50 includes ring-0 10, ring-1 20, ring-2 30, ring-3 40, and a processor nub loader 52. The processor nub loader 52 is an instance of a processor executive (PE) handler. The PE handler is used to handle and/or manage a processor executive (PE) as will be discussed later. The logical operating architecture 50 has two modes of operation: normal execution mode and isolated execution mode. Each ring in the logical operating architecture 50 can operate in both modes. The processor nub loader 52 operates only in the isolated execution mode.

Ring-0 10 includes two portions: a normal execution Ring-0 11 and an isolated execution Ring-0 15. The normal execution Ring-0 11 includes software modules that are critical for the operating system, usually referred to as kernel. These software modules include primary operating system (e.g., kernel) 12, software drivers 13, and hardware drivers 14. The isolated execution Ring-0 15 includes an operating system (OS) nub 16 and a processor nub 18. The OS nub 16 and the processor nub 18 are instances of an OS executive (OSE) and processor executive (PE), respectively. The OSE and the PE are part of executive entities that operate in a secure environment associated with the isolated area 70 and the isolated execution mode. The processor nub loader 52 is a protected bootstrap loader code held within a chipset in the system and is responsible for loading the processor nub 18 from the processor or chipset into an isolated area as will be explained later.

Similarly, ring-1 20, ring-2 30, and ring-3 40 include normal execution ring-1 21, ring-2 31, ring-3 41, and isolated execution ring-1 25, ring-2 35, and ring-3 45, respectively. In particular, normal execution ring-3 includes N applications 421 to 42N and isolated execution ring-3 includes K applets 461 to 46K.

One concept of the isolated execution architecture is the creation of an isolated region in the system memory, referred to as an isolated area, which is protected by both the processor and chipset in the computer system. The isolated region may also be in cache memory, protected by a translation lookaside buffer (TLB) access check. Access to this isolated region is permitted only from a front side bus (FSB) of the processor, using special bus (e.g., memory read and write) cycles, referred to as isolated read and write cycles. The special bus cycles are also used for snooping. The isolated read and write cycles are issued by the processor executing in an isolated execution mode. The isolated execution mode is initialized using a privileged instruction in the processor, combined with the processor nub loader 52. The processor nub loader 52 verifies and loads a ring-0 nub software module (e.g., processor nub 18) into the isolated area. The processor nub 18 provides hardware-related services for the isolated execution.

One task of the processor nub 18 is to verify and load the ring-0 OS nub 16 into the isolated area, and to generate the root of a key hierarchy unique to a combination of the platform, the processor nub 18, and the operating system nub 16. The operating system nub 16 provides links to services in the primary OS 12 (e.g., the unprotected segments of the operating system), provides page management within the isolated area, and has the responsibility for loading ring-3 application modules 45, including applets 461 to 46K, into protected pages allocated in the isolated area. The operating system nub 16 may also load ring-0 supporting modules.

The operating system nub 16 may choose to support paging of data between the isolated area and ordinary (e.g., non-isolated) memory. If so, then the operating system nub 16 is also responsible for encrypting and hashing the isolated area pages before evicting the page to the ordinary memory, and for checking the page contents upon restoration of the page. The isolated mode applets 46_1 to 46_K and their data are tamper- and monitor-proof from all software attacks from other applets, as well as from non-

isolated-space applications (e.g., 42_1 to 42_N), dynamic link libraries (DLLs), drivers and even the primary operating system 12. Only the processor nub 18 or the operating system nub 16 can interfere with or monitor the applet's execution.

Figure 1B is a diagram illustrating accessibility of various elements in the operating system 10 and the processor according to one embodiment of the invention. For illustration purposes, only elements of ring-0 10 and ring-3 40 are shown. The various elements in the logical operating architecture 50 access an accessible physical memory 60 according to their ring hierarchy and the execution mode.

The accessible physical memory 60 includes an isolated area 70 and a non-isolated area 80. The isolated area 70 includes applet pages 72 and nub pages 74. The non-isolated area 80 includes application pages 82 and operating system pages 84. The isolated area 70 is accessible only to elements of the operating system and processor operating in isolated execution mode. The non-isolated area 80 is accessible to all elements of the ring-0 operating system and to the processor.

The normal execution ring-0 11 including the primary OS 12, the software drivers 13, and the hardware drivers 14, can access both the OS pages 84 and the application pages 82. The normal execution ring-3, including applications 421 to 42N, can access only to the application pages 82. Both the normal execution ring-0 11 and ring-3 41, however, cannot access the isolated area 70.

The isolated execution ring-0 15, including the OS nub 16 and the processor nub 18, can access to both of the isolated area 70, including the applet pages 72 and the nub pages 74, and the non-isolated area 80, including the application pages 82 and the OS pages 84. The isolated execution ring-3 45, including applets 46_1 to 46_K, can access only to the application pages 82 and the applet pages 72. The applets 46_1 to 46_K reside in the isolated area 70.

Figure 1C is a diagram illustrating a computer system 100 in which one embodiment of the invention can be practiced. The computer system 100 includes a processor 110, a host bus 120, a memory controller hub (MCH) 130, a system memory 140, an input/output controller hub (ICH) 150, a non-volatile memory, or system flash, 160, a mass storage device 170, input/output devices 175, a token bus 180, a motherboard (MB) token 182, a reader 184, and a token 186. The MCH 130 may be integrated into a chipset that integrates multiple functionalities such as the isolated execution mode, host-to-peripheral bus interface, memory control. Similarly, the ICH 150 may also be integrated into a chipset together or separate from the MCH 130 to perform I/O functions. For clarity, not all the peripheral buses are shown. It is contemplated that the system 100 may also include peripheral buses such as Peripheral Component Interconnect (PCI), accelerated graphics port (AGP), Industry Standard Architecture (ISA) bus, and Universal Serial Bus (USB), etc.

The processor 110 represents a central processing unit of any type of architecture, such as complex instruction set computers (CISC), reduced instruction set computers (RISC), very long instruction word (VLIW), or hybrid architecture. In one embodiment, the processor 110 is compatible with an Intel Architecture (IA) processor, such as the Pentium™ series, the IA-32™ and the IA-64™. The processor 110 includes a normal execution mode 112 and an isolated execution circuit 115. The normal execution mode 112 is the mode in which the processor 110 operates in a non-secure environment, or a normal environment without the security features provided by the isolated execution mode. The isolated execution circuit 115 provides a mechanism to allow the processor 110 to operate in an isolated execution mode. The isolated execution circuit 115 provides hardware and software support for the isolated execution mode. This support includes configuration for isolated execution, definition of an isolated area, definition (e.g.,

decoding and execution) of isolated instructions, generation of isolated access bus cycles, and generation of isolated mode interrupts.

In one embodiment, the computer system 100 can be a single processor system, such as a desktop computer, which has only one main central processing unit, e.g. processor 110. In other embodiments, the computer system 100 can include multiple processors, e.g. processors 110, 110a, 110b, etc., as shown in Figure 1C. Thus, the computer system 100 can be a multi-processor computer system having any number of processors. For example, the multi-processor computer system 100 can operate as part of a server or workstation environment. The basic description and operation of processor 110 will be discussed in detail below. It will be appreciated by those skilled in the art that the basic description and operation of processor 110 applies to the other processors 110a and 110b, shown in Figure 1C, as well as any number of other processors that may be utilized in the multi-processor computer system 100 according to one embodiment of the present invention.

The processor 110 may also have multiple logical processors. A logical processor, sometimes referred to as a thread, is a functional unit within a physical processor having an architectural state and physical resources allocated according to some partitioning policy. Within the context of the present invention, the terms "thread" and "logical processor" are used to mean the same thing. A multi-threaded processor is a processor having multiple threads or multiple logical processors. A multi-processor system (e.g., the system comprising the processors 110, 110a, and 110b) may have multiple multi-threaded processors.

The host bus 120 provides interface signals to allow the processor 110 or processors 110, 100a, and 110b to communicate with other processors or devices, e.g., the MCH 130. In addition to normal mode, the host bus 120 provides an isolated access

bus mode with corresponding interface signals for memory read and write cycles when the processor 110 is configured in the isolated execution mode. The isolated access bus mode is asserted on memory accesses initiated while the processor 110 is in the isolated execution mode. The isolated access bus mode is also asserted on instruction pre-fetch and cache write-back cycles if the address is within the isolated area address range and the processor 110 is initialized in the isolated execution mode. The processor 110 responds to snoop cycles to a cached address within the isolated area address range if the isolated access bus cycle is asserted and the processor 110 is initialized into the isolated execution mode.

The MCH 130 provides control and configuration of memory and input/output devices such as the system memory 140 and the ICH 150. The MCH 130 provides interface circuits to recognize and service isolated access assertions on memory reference bus cycles, including isolated memory read and write cycles. In addition, the MCH 130 has memory range registers (e.g., base and length registers) to represent the isolated area in the system memory 140. Once configured, the MCH 130 aborts any access to the isolated area that does not have the isolated access bus mode asserted.

The system memory 140 stores system code and data. The system memory 140 is typically implemented with dynamic random access memory (DRAM) or static random access memory (SRAM). The system memory 140 includes the accessible physical memory 60 (shown in Figure 1B). The accessible physical memory includes a loaded operating system 142, the isolated area 70 (shown in Figure 1B), and an isolated control and status space 148. The loaded operating system 142 is the portion of the operating system that is loaded into the system memory 140. The loaded OS 142 is typically loaded from a mass storage device via some boot code in a boot storage such as a boot read only memory (ROM). The isolated area 70, as shown in Figure 1B, is the memory area that is defined by the processor 110 when operating in the isolated execution mode.

Access to the isolated area 70 is restricted and is enforced by the processor 110 and/or the MCH 130 or other chipset that integrates the isolated area functionalities. The isolated control and status space 148 is an input/output (I/O)-like, independent address space defined by the processor 110 and/or the MCH 130. The isolated control and status space 148 contains mainly the isolated execution control and status registers. The isolated control and status space 148 does not overlap any existing address space and is accessed using the isolated bus cycles. The system memory 140 may also include other programs or data which are not shown.

The ICH 150 represents a known single point in the system having the isolated execution functionality. For clarity, only one ICH 150 is shown. The system 100 may have many ICHs similar to the ICH 150. When there are multiple ICHs, a designated ICH is selected to control the isolated area configuration and status. In one embodiment, this selection is performed by an external strapping pin. As is known by one skilled in the art, other methods of selecting can be used, including using programmable configuring registers. The ICH 150 has a number of functionalities that are designed to support the isolated execution mode in addition to the traditional I/O functions. In particular, the ICH 150 includes an isolated bus cycle interface 152, the processor nub loader 52 (shown in Figure 1A), a digest memory 154, a cryptographic key storage 155, an isolated execution logical processor manager 156, and a token bus interface 159.

The isolated bus cycle interface 152 includes circuitry to interface to the isolated bus cycle signals to recognize and service isolated bus cycles, such as the isolated read and write bus cycles. The processor nub loader 52, as shown in Figure 1A, includes a processor nub loader code and its digest (e.g., hash) value. The processor nub loader 52 is invoked by execution of an appropriate isolated instruction (e.g., IsoCreate) and is transferred to the isolated area 70. From the isolated area 80, the processor nub loader 52 copies the processor nub 18 from the system flash memory (e.g., the processor nub code

18 in non-volatile memory 160) into the isolated area 70, verifies and logs its integrity, and manages a symmetric key used to protect the processor nub's secrets. In one embodiment, the processor nub loader 52 is implemented in read only memory (ROM). For security purposes, the processor nub loader 52 is unchanging, tamper-proof and non-substitutable. The digest memory 154, typically implemented in RAM, stores the digest (e.g., hash) values of the loaded processor nub 18, the operating system nub 16, and any other critical modules (e.g., ring-0 modules) loaded into the isolated execution space. The cryptographic key storage 155 holds a symmetric encryption/decryption key that is unique for the platform of the system 100. In one embodiment, the cryptographic key storage 155 includes internal fuses that are programmed at manufacturing. Alternatively, the cryptographic key storage 155 may also be created with a random number generator and a strap of a pin. The isolated execution logical processor manager 156 manages the operation of logical processors operating in isolated execution mode. In one embodiment, the isolated execution logical processor manager 156 includes a logical processor count register that tracks the number of logical processors participating in the isolated execution mode. The token bus interface 159 interfaces to the token bus 180. A combination of the processor nub loader digest, the processor nub digest, the operating system nub digest, and optionally additional digests, represents the overall isolated execution digest, referred to as isolated digest. The isolated digest is a fingerprint identifying the ring-0 code controlling the isolated execution configuration and operation. The isolated digest is used to attest or prove the state of the current isolated execution.

The non-volatile memory 160 stores non-volatile information. Typically, the non-volatile memory 160 is implemented in flash memory. The non-volatile memory 160 includes the processor nub 18. The processor nub 18 provides the initial set-up and low-level management of the isolated area 70 (in the system memory 140), including verification, loading, and logging of the operating system nub 16, and the management of

the symmetric key used to protect the operating system nub's secrets. The processor nub 18 may also provide application programming interface (API) abstractions to low-level security services provided by other hardware. The processor nub 18 may also be distributed by the original equipment manufacturer (OEM) or operating system vendor (OSV) via a boot disk.

The mass storage device 170 stores archive information such as code (e.g., processor nub 18), programs, files, data, applications (e.g., applications 42_1 to 42_N), applets (e.g., applets 46_1 to 46_K) and operating systems. The mass storage device 170 may include compact disk (CD) ROM 172, floppy diskettes 174, and hard drive 176, and any other magnetic or optical storage devices. The mass storage device 170 provides a mechanism to read machine-readable media. When implemented in software, the elements of the present invention are the code segments to perform the necessary tasks. The program or code segments can be stored in a processor readable medium or transmitted by a computer data signal embodied in a carrier wave, or a signal modulated by a carrier, over a transmission medium. The "processor readable medium" may include any medium that can store or transfer information. Examples of the processor readable medium include an electronic circuit, a semiconductor memory device, a ROM, a flash memory, an erasable programmable ROM (EPROM), a floppy diskette, a compact disk CD-ROM, an optical disk, a hard disk, a fiber optical medium, a radio frequency (RF) link, etc. The computer data signal may include any signal that can propagate over a transmission medium such as electronic network channels, optical fibers, air, electromagnetic, RF links, etc. The code segments may be downloaded via computer networks such as the Internet, an Intranet, etc.

I/O devices 175 may include any I/O devices to perform I/O functions. Examples of I/O devices 175 include a controller for input devices (e.g., keyboard, mouse, trackball,

pointing device), media card (e.g., audio, video, graphics), a network card, and any other peripheral controllers.

The token bus 180 provides an interface between the ICH 150 and various tokens in the system. A token is a device that performs dedicated input/output functions with security functionalities. A token has characteristics similar to a smart card, including at least one reserved-purpose public/private key pair and the ability to sign data with the private key. Examples of tokens connected to the token bus 180 include a motherboard token 182, a token reader 184, and other portable tokens 186 (e.g., smart card). The token bus interface 159 in the ICH 150 connects through the token bus 180 to the ICH 150 and ensures that when commanded to prove the state of the isolated execution, the corresponding token (e.g., the motherboard token 182, the token 186) signs only valid isolated digest information. For purposes of security, the token should be connected to the digest memory.

GENERATING ISOLATED BUS CYLCLES

The isolated execution mode protects the platform, system, or environment from attacks, especially attacks by software. In one embodiment of the present invention, a mechanism is provided to prevent software attacks. This mechanism includes generation of special bus cycles by the processor. The special bus cycle is used to enable the processor to perform operations such that they could not be performed by software instructions.

A significant feature of the isolated execution mode is that it prevents a corrupted ring-0 area from getting access to a user's important data. It does this by using execution mode. In this manner, both ring-0 programs and direct memory access (DMA) devices are restricted from accessing the isolated memory area.

Figure 2A is a diagram illustrating the isolated execution circuit 115 shown in Figure 1C according to one embodiment of the invention. The isolated execution circuit 115 includes an instruction decoder and execution unit 210, a translation lookaside buffer (TLB) 218, and an isolated bus cycle generator 220.

The instruction decoder and execution unit 210 receives an instruction stream 215 from an instruction fetch unit. The instruction stream 215 includes a number of instructions. The instruction decoder and execution unit 210 decodes the instructions and executes the decoded instructions. These instructions may be at the micro- or macro-level. The instruction decoder and execution unit 210 may be a physical circuit or an abstraction of a process of decoding and execution of instructions. In addition, the instructions may include isolated instructions and non-isolated instructions. The instruction decoder and execution unit 210 generates a virtual address 212 when there is an access transaction. The TLB 218 translates the virtual address 212 into a physical address which is part of access information 226. The instruction decoeder and execution unit 210 interfaces with the isolated bus cycle generator 220 via control/status information 222 and operand 224. The control/status information 222 includes control bits to manipulate various elements in the isolated bus cycle generator 220 and status data from the isolated bus cycle generator 220. The operand 224 includes data to be written to and read from the isolated bus cycle generator 220. The access information 226 includes address, read/write, and access type information.

The isolated bus cycle generator 220 receives and provides the control/status information 222, the operand 224, and receives the access information 226 from the instruction decoder and execution unit 210 as a result of instruction execution. The isolated bus cycle generator 220 generates an isolated bus cycle 230. The isolated bus cycle 230 includes information sent to devices (e.g., chipsets) external to the processor 110 to indicate that the processor 110 is executing an isolated mode instruction. The

isolated bus cycle 230 may also be used internally by the processor 110 to control and monitor other isolated or non-isolated activities.

Figure 2B is a diagram illustrating the isolated bus cycle generator 220 shown in Figure 2A according to one embodiment of the invention. The isolated bus cycle generator 220 includes a configuration storage 250, an access generator circuit 270, and a bus cycle decoder 280. The isolated bus cycle generator 220 exchanges operand 224 with and receives the access information 226 from the instruction decoder and execution unit 210 shown in Figure 2A. The access information 226 includes a physical address 282, a read/write (RD/WR#) signal 284 and an access type 286. The access information 226 is generated during an access transaction by the processor 110.

The configuration storage 250 contains configuration parameters to configure the processor 110 in one of a normal execution mode and an isolated execution mode. The configuration storage 250 receives the operand 224 from the instruction decoder and execution unit 210 (Figure 2A) and includes a processor control register 252 and an isolated setting 260. The processor control register 252 contains an execution mode word 253. The execution mode word 253 is asserted when the processor 110 is configured in the isolated execution mode. In one embodiment, the execution mode word 253 is a single bit indicating if the processor 110 is in the isolated execution mode. The isolated setting 260 defines the isolated memory area (e.g., the isolated area 70 in the system memory 140 shown in Figure 1C). The isolated setting 260 may include a mask register 262, a base register 264, and a length register 266. The mask register 262 contains an isolated mask value 263. The base register 264 contains an isolated base value 265. The length register 266 contains a length value 267. The isolated mask, base, and length values 263, 265, and 267 are used to define the isolated memory area. The isolated memory area may be defined by using any combination of the mask, base, and length values 263, 265, and 267. For example, the base value 265 corresponds to the starting

address of the isolated memory area, while the sum of the base value 265 and the length value 267 corresponds to the ending address of the isolated memory area.

The access generator circuit 270 generates an isolated access signal 272 using at least one of the isolated area parameters in the configuration storage 250 and the access information 226 in a transaction generated by the processor 110. The isolated access signal 272 is asserted when the processor 110 is configured in the isolated execution mode. The access generator circuit 270 receives the physical address 282 and the read/write signal 284. The access generator circuit 270 also generates an access grant signal 274 to indicate if an isolated access has been granted. A similar checking is also performed for snoop checking when the physical address is provided at the FSB.

The bus cycle decoder 280 generates an isolated bus cycle 230 corresponding to a destination in the transaction using the asserted isolated access signal 272 and the access information 226. The bus cycle decoder 280 receives the read/write signal 284 and the access type 286. The read/write signal 284 indicates whether a read or a write transaction is being performed. The access type 286 indicates a type of access, including a memory reference, an input/output (I/O) reference, a logical processor entry to an isolated enabled state, and a logical processor withdrawal from an isolated enabled state. The destination of the transaction may be one of an isolated memory area, an isolated register, and an isolated state, corresponding to the memory reference, the I/O reference, and the isolated enabled state, respectively.

Figure 3 is a diagram illustrating the access generator circuit 270 shown in Figure 2B according to one embodiment of the invention. The access generator circuit 270 includes an address detector 310 and an access grant generator 320.

The address detector 310 receives the isolated setting 260 (e.g., the isolated mask value 263, the isolated base value 265, the isolated length value 267) from the

configuration storage 250 in Figure 2B. The address detector 310 detects if the physical address 282 is within the isolated memory area defined by the isolated setting 260. In one embodiment, the isolated memory area is defined by the isolated mask and base values 263 and 265. The address detector 310 includes a masking element 312 and a comparator 314. The masking element 312 masks the physical address 282 with the isolated mask value 263. In one embodiment, the masking element 312 performs a logical AND operation. The comparator 314 compares the result of the masking operation done by the masking element 312 and the isolated base value 265, and generates the isolated access signal 272. The isolated access signal 272 is asserted when the physical address 282 is within the isolated memory area as defined by the isolated mask and base values 263 and 265, respectively.

The access grant generator 320 combines the isolated access signal 272 and the execution mode word 253 to generate an access grant signal 274. The access grant signal 274 is asserted when both the isolated access signal 272 and the execution mode word 253 are asserted to indicate that an isolated access is valid or allowed as configured. In one embodiment, the access grant generator 320 performs a logical AND operation.

Figure 4 is a diagram illustrating the bus cycle decoder 280 according to one embodiment of the invention. The bus cycle decoder 280 includes a decoder 410 which implements a truth table 420.

The decoder 410 receives the access information 226 including access type 286 and the read/write signal 284, and the isolated access signal 272 from the access generator 270 (Figure 2B). The decoder 410 may be implemented with a hardwired logic circuit or a programmable logic device using the truth table 420. The decoder 410 generates the isolated bus cycle 230 with appropriates coded information.

The truth table 420 provides logic equations to generate the isolated bus cycle 230. The isolated bus cycle 230 may be coded in a number of ways. In one embodiment, the isolated bus cycle 230 is coded using binary information. There are essentially seven values or states of the isolated bus cycle 230: a not-available cycle, an isolated logical processor entry cycle, an isolated logical processor withdrawal cycle, an isolated data read cycle, an isolated data write cycle, an isolated control read cycle, and an isolated control write cycle. The isolated bus cycle 230 can be coded using 3 bits of data. The isolated bus cycle 230 can be made available to the external devices so that it can be decoded and used accordingly. The isolated logical processor entry and withdrawal cycles form a logical processor cycle. The isolated data read and write cycles form a data access cycle. The isolated control read and write cycles form a control access cycle. The isolated logical processor cycles are typically generated when the processor operates in ring-0. The isolated data access cycles are typically generated when the processor accesses an isolated memory area. The data access information can be decoded with the normal bus protocol. The control access cycles are typically generated when the processor accesses an isolated register residing in the MCH and the ICH in ring-0.

The truth table 420 includes as inputs the access type 286, the read/write (RD/WR#) 284, and the isolated access signal 272. The access type 286 includes an isolated enabled state and a memory/input-output (M/IO#). The isolated enabled state has two states: an enabled entry state and an enabled withdrawal state. The enabled entry state is asserted when a logical processor enters an isolated execution mode. This occurs when the processor executes an appropriate isolated instruction such as an isolated initialize (iso_init) instruction. The enabled withdrawal state is asserted when a logical processor withdraws from an isolated execution mode. This occurs when the processor executes an appropriate isolated instruction such as an isolated close (iso_close)

instruction. In addition, the truth table 420 may also include the ring level to restrict access. For example, the isolated input/output space is accessible only at ring-0.

The isolated bus cycle 230 is not available when the isolated access signal 272 is de-asserted. The isolated bus cycle 230 is the logical processor entry cycle when the isolated access signal 272 is asserted and the isolated enabled state is in the enabled entry state. The isolated bus cycle 230 is the logical processor withdrawal cycle when the isolated access signal 272 is asserted and the isolated enabled state is in the enabled withdrawal state. The isolated bus cycle 230 is the isolated data read cycle when the isolated access signal 272 is asserted, the M/IO# indicates a memory reference (e.g., M/IO# = 1), and the read/write indicates a read operation (e.g., RD/WR# = 1). The isolated bus cycle 230 is the isolated data write cycle when the isolated access signal 272 is asserted, the M/IO# indicates a memory reference (e.g., M/IO# = 1), and the read/write indicates a write operation (e.g., RD/WR# = 0). The isolated bus cycle 230 is the isolated control read cycle when the isolated access signal 272 is asserted, the M/IO# indicates an input/output reference (e.g., M/IO# = 0), and the read/write indicates a read operation (e.g., RD/WR# = 1). The isolated bus cycle 230 is the isolated control write cycle when the isolated access signal 272 is asserted, the M/IO# indicates an input/output reference (e.g., M/IO# = 0), and the read/write indicates a write operation (e.g., RD/WR# = 0).

Figure 5 is a flowchart illustrating a process 500 to generate isolated bus cycles for isolated execution according to one embodiment of the invention.

Upon START, the process 500 defines an isolated memory area using the isolated setting (e.g., isolated mask and base values) (Block 510). Then, the process 500 asserts the execution mode word in the processor control register to configure the processor in the isolated execution mode (Block 520). Next, the process 500 determines if the physical address as generated in a transaction is within the isolated memory area as

defined by the isolated setting (Block 530). If not, the process 500 generates a failure or fault condition or performs access to the non-isolated memory area if allowed (Block 535) and is then terminated. Otherwise, the process 500 asserts the isolated access signal (Block 540).

Next, the process 500 determines what access type is being used in the transaction (Block 550). If the access type is a memory reference, the process 500 generates a data access cycle (Block 560). If the access type is an input/output reference, the process 500 generates a control access cycle (Block 570). If the access type is an entry or withdrawal state, the process 500 generates a logical processor access cycle (Block 580). Then the process 500 is terminated.

In summary, the present invention is a method and apparatus to generate an isolated bus cycle for a transaction in a processor. A configuration storage contains configuration parameters to configure a processor in one of a normal execution mode and an isolated execution mode. An access generator circuit generates an isolated access signal using at least one of the isolated area parameters and access information in the transaction. The isolated access signal is asserted when the processor is configured in the isolated execution mode. A bus cycle decoder generates an isolated bus cycle corresponding to a destination in the transaction using the asserted isolated access signal and the access information.

In one embodiment, the configuration parameters include an isolated setting and an execution mode word. The destination in the transaction may be one of an isolated memory area, an isolated register, and an isolated state. The isolated memory area is located in a memory external to the processor. The isolated register may be located in a chipset external to the processor. The access information includes a physical address and an access type. The configuration storage includes a register to contain the isolated setting

for defining the isolated memory area. The isolated setting may be a mask value, a base value, a length value, or any of their combinations. The configuration storage further includes a processor control register to contain the execution mode word which is asserted when the processor is configured in the isolated execution mode. The access generator circuit includes an address detector to detect if the physical address is within the isolated memory area defined by the isolated setting. The isolated bus cycle may be one of a data access cycle, a control access cycle, and a logical processor access cycle. The data access cycle is generated when the access type is a memory reference to the isolated memory area. The control access cycle is generated when the access type is an input/output reference to the isolated register which may be external to the processor. The logical processor access cycle is generated when the access type is one of a logical processor entry to and a logical processor withdrawal from the isolated state. The logical processor entry to the isolated state updates a logical processor counter in the chipset in a first direction (e.g., increment). The logical processor withdrawal from the isolated state updates a logical processor counter in the chipset in a second direction (e.g., decrement).

While this invention has been described with reference to illustrative embodiments, this description is not intended to be construed in a limiting sense. Various modifications of the illustrative embodiments, as well as other embodiments of the invention, which are apparent to persons skilled in the art to which the invention pertains are deemed to lie within the spirit and scope of the invention.

CLAIMS

What is claimed is:

1. An apparatus comprising:

a configuration storage containing configuration parameters to configure a processor in one of a normal execution mode and an isolated execution mode;

an access generator circuit coupled to the configuration storage to generate an isolated access signal using at least one of the configuration parameters and access information in a transaction, the isolated access signal being asserted when the processor is configured in the isolated execution mode; and

a bus cycle decoder coupled to the access generator circuit to generate an isolated bus cycle corresponding to a destination in the transaction using the asserted isolated access signal and the access information.

2. The apparatus of claim 1 wherein the configuration parameters include an isolated setting and an execution mode word.

3. The apparatus of claim 1 wherein the destination in the transaction is one of an isolated memory area in a memory external to the processor, an isolated register, and an isolated state.

4. The apparatus of claim 3 wherein the access information comprises a physical address and an access type.

5. The apparatus of claim 4 wherein the configuration storage comprises:

a register to contain the isolated setting for defining the isolated memory area.

6. The apparatus of claim 5 wherein the isolated setting is one of a mask value, a base value, and a length value.

7. The apparatus of claim 6 wherein the configuration storage further comprises:

a processor control register to contain the execution mode word, the execution mode word being asserted when the processor is configured in the isolated execution mode.

8. The apparatus of claim 7 wherein the access generator circuit comprises:

an address detector to detect if the physical address is within the isolated memory area defined by the isolated setting.

9. The apparatus of claim 8 wherein the isolated bus cycle is one of a data access cycle, a control access cycle, and a logical processor access cycle.

10. The apparatus of claim 9 wherein the data access cycle is generated when the access type is a memory reference to the isolated memory area.

11. The apparatus of claim 9 wherein the isolated register is in a chipset external to the processor.

12. The apparatus of claim 11 wherein the control access cycle is generated when the access type is an input/output reference to the isolated register.

13. The apparatus of claim 9 wherein the logical processor access cycle is generated when the access type is one of a logical processor entry to and a logical processor withdrawal from the isolated state.

14. The apparatus of claim 13 wherein the logical processor entry to the isolated state updates a logical processor counter in the chipset in a first direction.

15. The apparatus of claim 13 wherein the logical processor withdrawal from the isolated state updates a logical processor counter in the chipset in a second direction.

16. A method comprising:

configuring a processor in one of a normal execution mode and an isolated execution mode using a configuration storage in the processor, the configuration storage containing configuration parameters;

asserting an isolated access signal by an access generator circuit using at least one of the isolated area parameters and access information in a transaction when the processor is configured in the isolated execution mode; and

generating an isolated bus cycle corresponding to a destination in the transaction by a bus cycle decoder using the asserted isolated access signal and the access information.

17. The method of claim 16 wherein the configuration parameters include an isolated setting and an execution mode word.

18. The method of claim 16 wherein the destination in the transaction is one of an isolated memory area in a memory external to the processor, an isolated register, and an isolated state.

19. The method of claim 18 wherein the access information includes a physical address and an access type.

20. The method of claim 19 wherein configuring comprises:

defining the isolated memory area using the isolated setting contained in a register.

21. The method of claim 20 wherein the isolated setting is one of a mask value a base value, and a length value.

22. The method of claim 21 wherein configuring further comprises:

asserting the execution mode word in a processor control register when the processor is configured in the isolated execution mode.

23. The method of claim 22 wherein asserting the isolated access signal comprises:

detecting if the physical address is within the isolated memory area defined by the isolated setting by an address detector.

24. The method of claim 23 wherein the isolated bus cycle is one of a data access cycle, a control access cycle, and a logical processor access cycle.

25. The method of claim 24 wherein the data access cycle is generated when the access type is a memory reference to the isolated memory area.

26. The method of claim 24 wherein the isolated register is in a chipset external to the processor.

27. The method of claim 26 wherein the control access cycle is generated when the access type is an input/output reference to the isolated register.

28. The method of claim 24 wherein the logical processor access cycle is generated when the access type is one of a logical processor entry to and a logical processor withdrawal from the isolated state.

29. The method of claim 28 wherein the logical processor entry to the isolated state updates a logical processor counter in the chipset in a first direction.

30. The method of claim 28 wherein the logical processor withdrawal from the isolated state updates a logical processor counter in the chipset in a second direction.

31. A system comprising:

a chipset;

a memory coupled to the chipset having an isolated memory area; and

a processor coupled to the chipset and the memory having an isolated bus cycle generator, the isolated bus cycle generator comprising:

a configuration storage containing configuration parameters to configure the processor in one of a normal execution mode and an isolated execution mode,

an access generator circuit coupled to the configuration storage to generate an isolated access signal using at least one of the isolated area parameters and access information in a transaction, the isolated access signal being asserted when the processor is configured in the isolated execution mode, and

a bus cycle decoder coupled to the access generator circuit to generate an isolated bus cycle corresponding to a destination in the transaction using the asserted isolated access signal and the access information.

32. The system of claim 31 wherein the configuration parameters include an isolated mode value and an execution mode word.

33. The system of claim 31 wherein the destination in the transaction is one of the isolated memory area, an isolated register, and an isolated state.

34. The system of claim 33 wherein the access information includes a physical address and an access type.

35. The system of claim 34 wherein the configuration storage comprises:

a register to contain the isolated setting for defining the isolated memory area.

36. The system of claim 35 wherein the isolated setting is one of a mask value, a base value, and a length value.

37. The system of claim 36 wherein the configuration storage further comprises:

a processor control register to contain the execution mode word, the execution mode word being asserted when the processor is configured in the isolated execution mode.

38. The system of claim 37 wherein the access generator circuit comprises:

an address detector to detect if the physical address is within the isolated memory area defined by the isolated setting.

39. The system of claim 38 wherein the isolated bus cycle is one of a data access cycle, a control access cycle, and a logical processor access cycle.

40. The system of claim 39 wherein the data access cycle is generated when the access type is a memory reference to the isolated memory area.

41. The system of claim 39 wherein the isolated register is in a chipset external to the processor.

42. The system of claim 41 wherein the control access cycle is generated when the access type is an input/output reference to the isolated register.

43. The system of claim 39 wherein the logical processor access cycle is generated when the access type is one of a logical processor entry to and a logical processor withdrawal from the isolated state.

44. The system of claim 43 wherein the logical processor entry to the isolated state updates a logical processor counter in the chipset in a first direction.

45. The system of claim 43 wherein the logical processor withdrawal from the isolated state updates a logical processor counter in the chipset in a second direction.

ABSTRACT OF THE DISCLOSURE

The present invention is a method and apparatus to generates an isolated bus cycle for a transaction in a processor. A configuration storage contains configuration parameters to configure a processor in one of a normal execution mode and an isolated execution mode. An access generator circuit generates an isolated access signal using at least one of the isolated area parameters and access information in the transaction. The isolated access signal is asserted when the processor is configured in the isolated execution mode. A bus cycle decoder generates an isolated bus cycle corresponding to a destination in the transaction using the asserted isolated access signal and the access information.

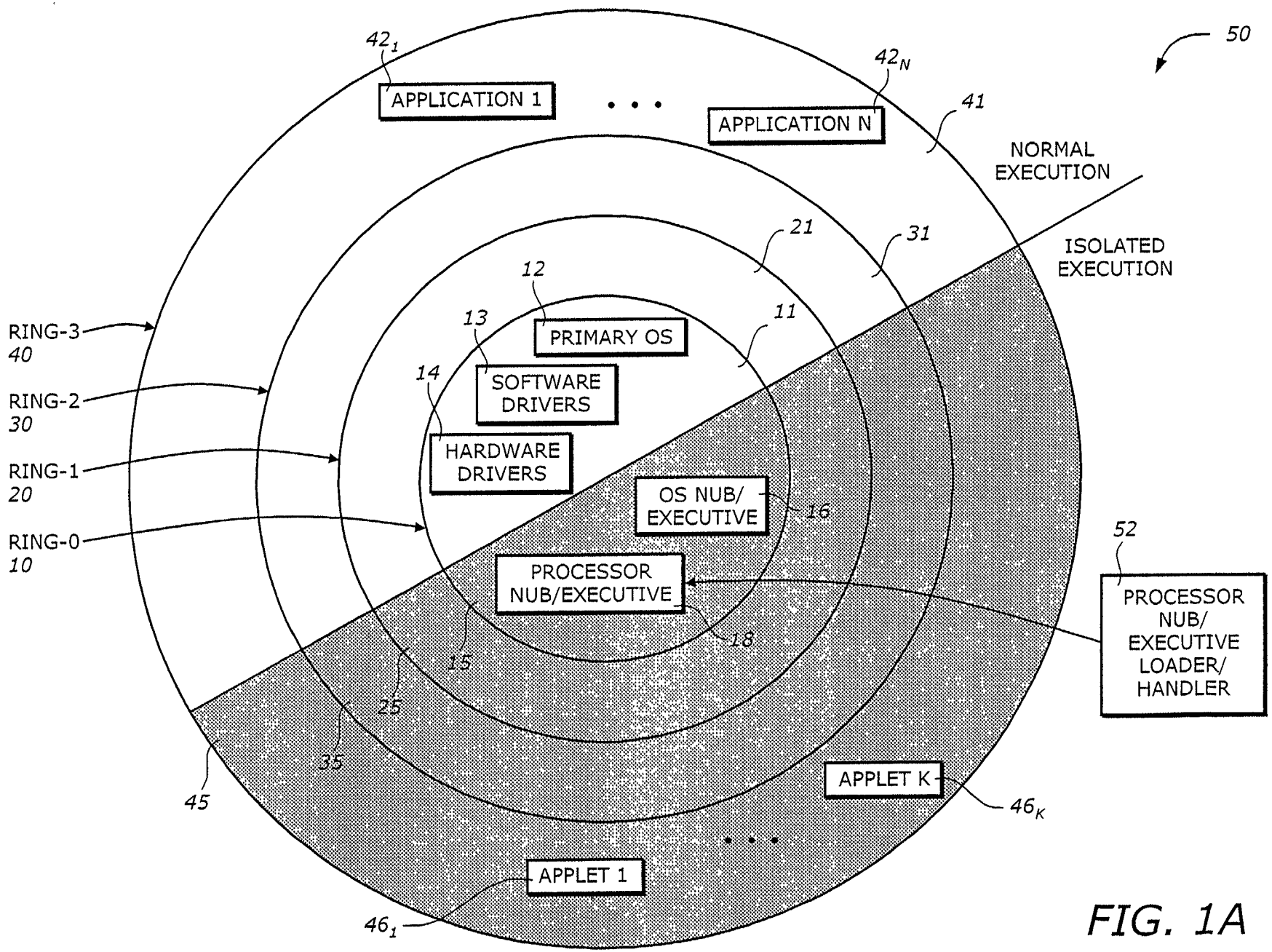

50
42₁
APPLICATION 1
42N
APPLICATION N
41
NORMAL EXECUTION
ISOLATED EXECUTION
21
31
12
PRIMARY OS
13
SOFTWARE DRIVERS
14
HARDWARE DRIVERS
11
RING-3
40
RING-2
30
RING-1
20
RING-0
10
OS NUB/ EXECUTIVE
16
PROCESSOR NUB/EXECUTIVE
18
15
25
35
45
52
PROCESSOR NUB/ EXECUTIVE LOADER/ HANDLER
APPLET K
46K
APPLET 1
46₁

FIG. 1A

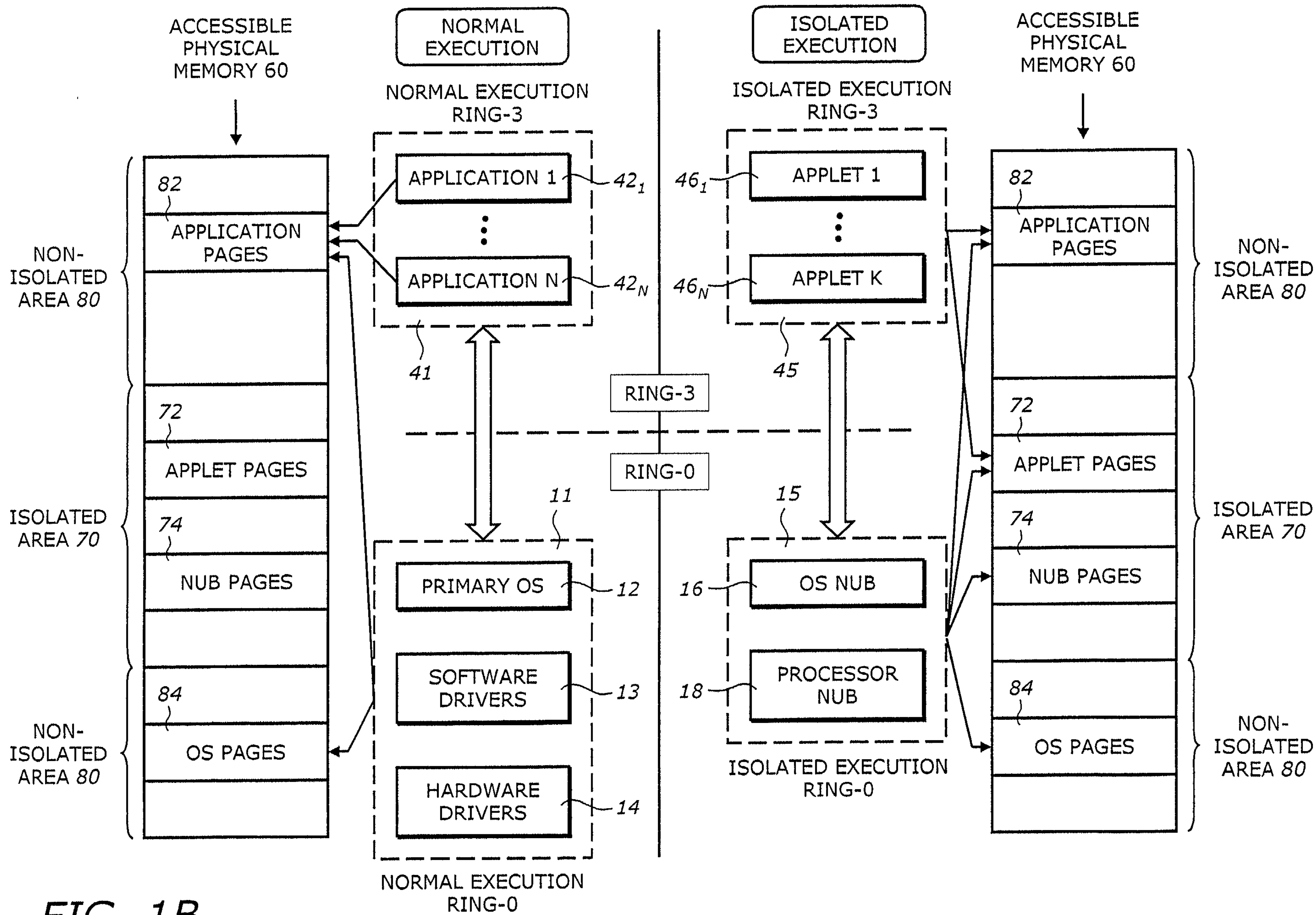

ACCESSIBLE PHYSICAL MEMORY 60
NORMAL EXECUTION
NORMAL EXECUTION RING-3
ISOLATED EXECUTION
ISOLATED EXECUTION RING-3
ACCESSIBLE PHYSICAL MEMORY 60
82
APPLICATION PAGES
72
APPLET PAGES
74
NUB PAGES
84
OS PAGES
NON-ISOLATED AREA 80
ISOLATED AREA 70
NON-ISOLATED AREA 80
APPLICATION 1
APPLICATION N
42_1
42_N
41
APPLET 1
APPLET K
46_1
46_N
45
RING-3
RING-0
11
PRIMARY OS
12
SOFTWARE DRIVERS
13
HARDWARE DRIVERS
14
NORMAL EXECUTION RING-0
15
OS NUB
16
PROCESSOR NUB
18
ISOLATED EXECUTION RING-0

FIG. 1B

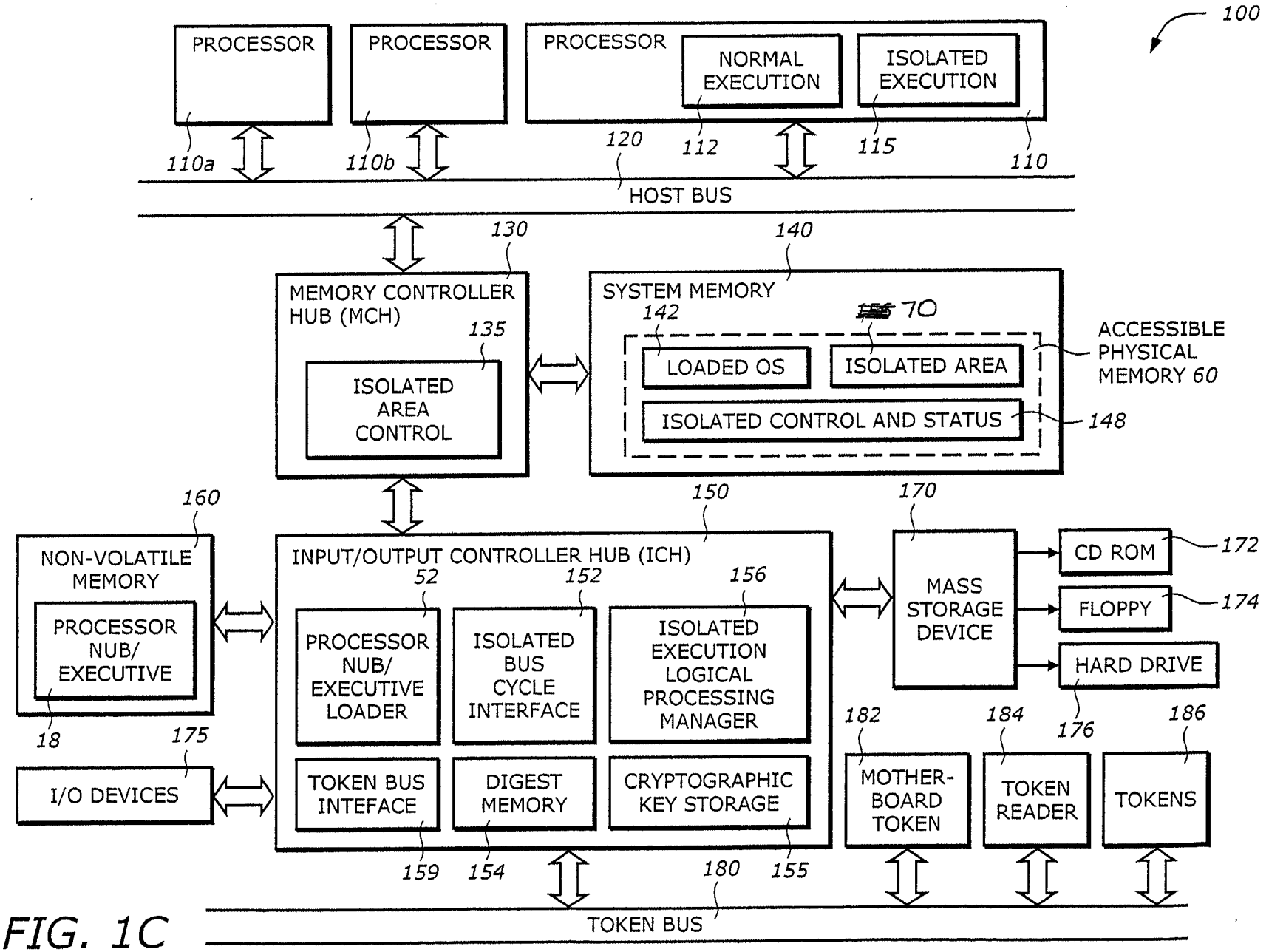

100
PROCESSOR
PROCESSOR
PROCESSOR
NORMAL EXECUTION
ISOLATED EXECUTION
110a
110b
120
112
115
110
HOST BUS
130
140
MEMORY CONTROLLER HUB (MCH)
135
ISOLATED AREA CONTROL
SYSTEM MEMORY
142
70
LOADED OS
ISOLATED AREA
ISOLATED CONTROL AND STATUS
148
ACCESSIBLE PHYSICAL MEMORY 60
160
150
170
NON-VOLATILE MEMORY
PROCESSOR NUB/ EXECUTIVE
18
175
INPUT/OUTPUT CONTROLLER HUB (ICH)
52
152
156
PROCESSOR NUB/ EXECUTIVE LOADER
ISOLATED BUS CYCLE INTERFACE
ISOLATED EXECUTION LOGICAL PROCESSING MANAGER
MASS STORAGE DEVICE
CD ROM
172
FLOPPY
174
HARD DRIVE
176
I/O DEVICES
TOKEN BUS INTEFACE
DIGEST MEMORY
CRYPTOGRAPHIC KEY STORAGE
182
184
186
MOTHER-BOARD TOKEN
TOKEN READER
TOKENS
159
154
180
155
TOKEN BUS

FIG. 1C

DOCKET: 1568.2560

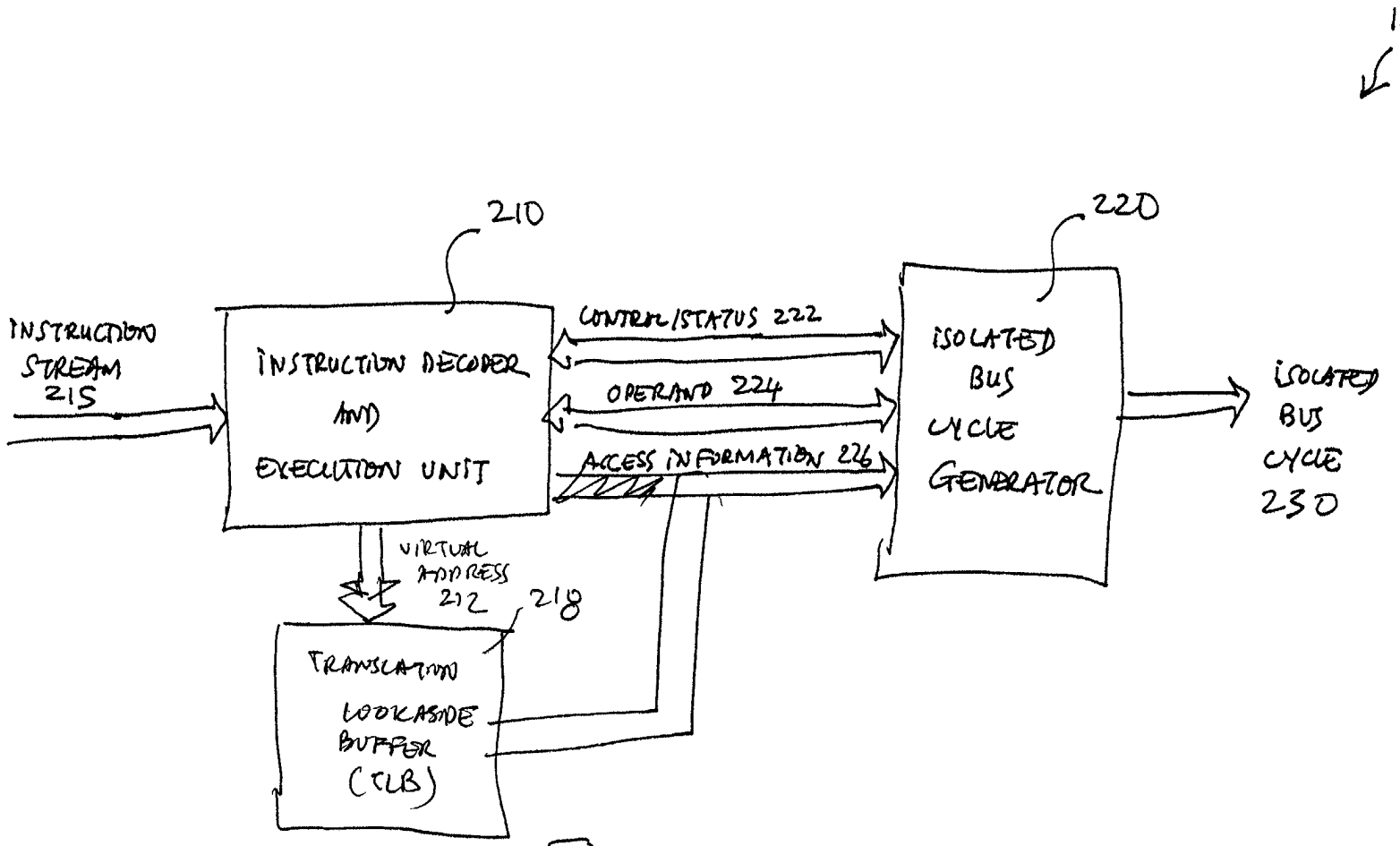


FIG. 2A

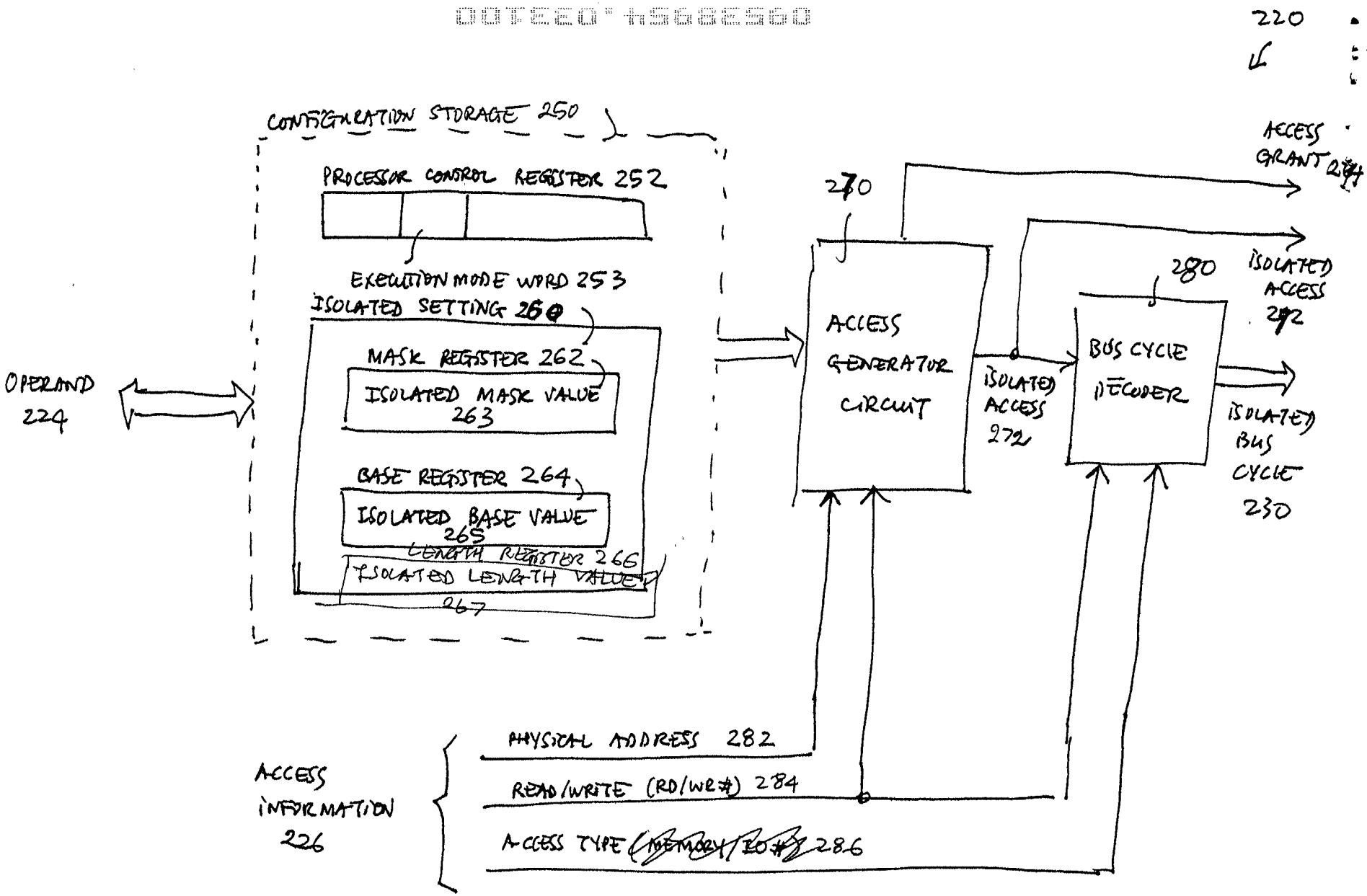

220
CONFIGURATION STORAGE 250
PROCESSOR CONTROL REGISTER 252
EXECUTION MODE WORD 253
ISOLATED SETTING 260
MASK REGISTER 262
ISOLATED MASK VALUE 263
BASE REGISTER 264
ISOLATED BASE VALUE 265
LENGTH REGISTER 266
ISOLATED LENGTH VALUE 267
OPERAND 224
270
ACCESS GENERATOR CIRCUIT
ACCESS GRANT 274
ISOLATED ACCESS 272
280
BUS CYCLE DECODER
ISOLATED ACCESS 272
ISOLATED BUS CYCLE 230
ACCESS INFORMATION 226
PHYSICAL ADDRESS 282
READ/WRITE (RD/WR#) 284
ACCESS TYPE (MEMORY/IO#) 286

FIG. 2B

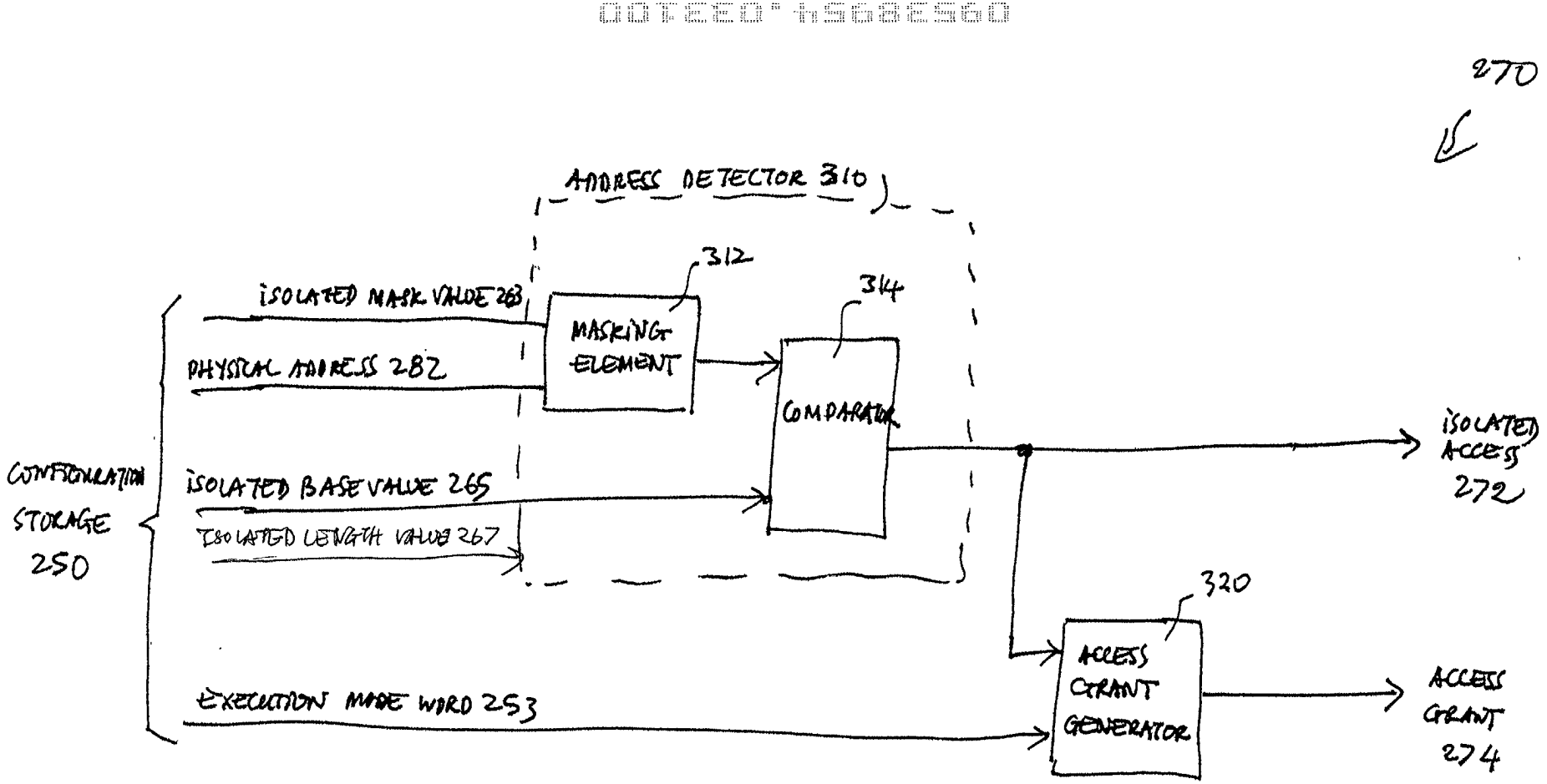
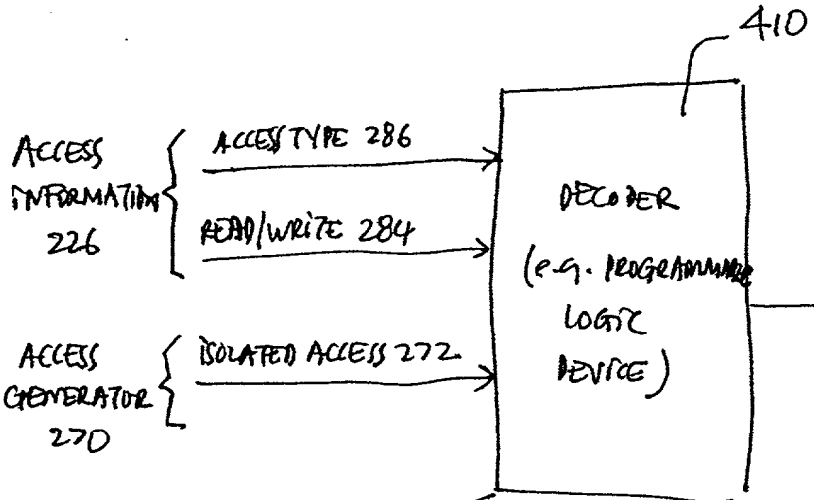

270
ADDRESS DETECTOR 310
312
314
ISOLATED MASK VALUE 263
MASKING ELEMENT
PHYSICAL ADDRESS 282
COMPARATOR
ISOLATED ACCESS 272
CONFIGURATION STORAGE 250
ISOLATED BASE VALUE 265
ISOLATED LENGTH VALUE 267
320
ACCESS GRANT GENERATOR
EXECUTION MODE WORD 253
ACCESS GRANT 274

FIG. 3



TRUTH TABLE 420

ACCESS TYPE			ISOLATED ACCESS	ISOLATED BUS CYCLE		RING LEVEL
ISOLATED ENABLED STATE	M/IO#	RD/WR#				
X	X	X	DE-ASSERTED	NOT AVAILABLE		0
ENTRY (e.g., ISO_INIT)	X	X	ASSERTED	THREAD ENTRY	THREAD CYCLE	0
WITHDRAW (e.g., ISO_CLOSE)	X	X	ASSERTED	THREAD WITHDRAWAL		0
X	MEMORY REFERENCE	READ	ASSERTED	ISOLATED DATA READ	DATA ACCESS CYCLE	0
X	MEMORY REFERENCE	WRITE	ASSERTED	ISOLATED DATA WRITE		0
X	I/O REFERENCE	READ	ASSERTED	ISOLATED CONTROL READ	CONTROL ACCESS CYCLE	0
X	I/O REFERENCE	WRITE	ASSERTED	ISOLATED CONTROL WRITE		0

FIG. 4

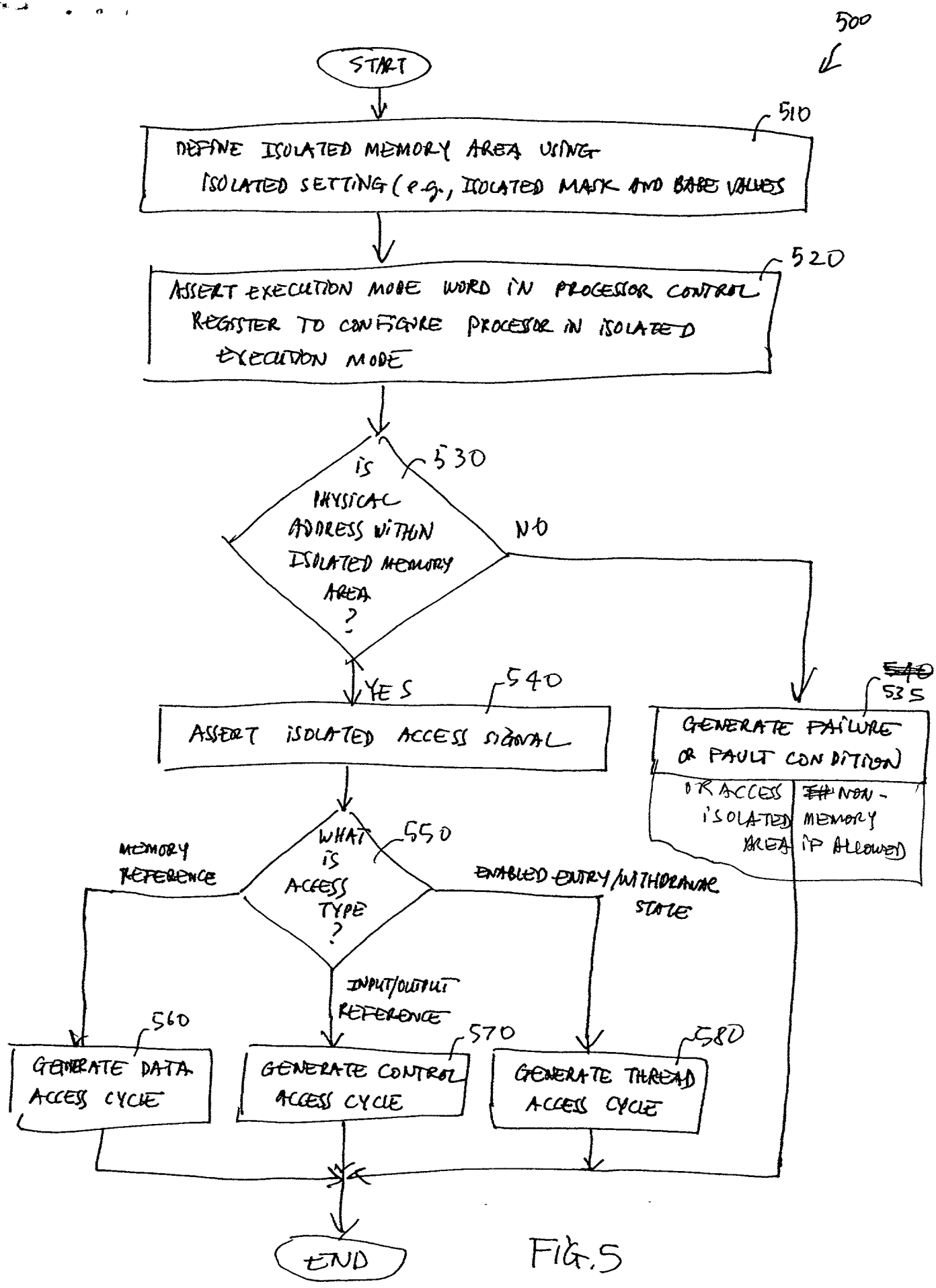

500
START
510
DEFINE ISOLATED MEMORY AREA USING ISOLATED SETTING (e.g., ISOLATED MASK AND BASE VALUES
520
ASSERT EXECUTION MODE WORD IN PROCESSOR CONTROL REGISTER TO CONFIGURE PROCESSOR IN ISOLATED EXECUTION MODE
530
IS PHYSICAL ADDRESS WITHIN ISOLATED MEMORY AREA ?
NO
YES
540
ASSERT ISOLATED ACCESS SIGNAL
535
GENERATE FAILURE OR FAULT CONDITION
OR ACCESS ISOLATED MEMORY AREA IF NON-MEMORY IF ALLOWED
550
WHAT IS ACCESS TYPE ?
MEMORY REFERENCE
INPUT/OUTPUT REFERENCE
ENABLED ENTRY/WITHDRAWAL STATE
560
GENERATE DATA ACCESS CYCLE
570
GENERATE CONTROL ACCESS CYCLE
580
GENERATE THREAD ACCESS CYCLE
END

FIG. 5

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION (FOR INTEL CORPORATION PATENT APPLICATIONS)

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below, next to my name.

I believe I am the original, first, and sole inventor (if only one name is listed below) or an original, first, and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

GENERATING ISOLATED BUS CYCLES FOR ISOLATED EXECUTION

the specification of which

[X] is attached hereto.
[ ] was filed on _____ as
United States Application Number _____
or PCT International Application Number _____
and was amended on _____ .
(if applicable)

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claim(s), as amended by any amendment referred to above. I do not know and do not believe that the claimed invention was ever known or used in the United States of America before my invention thereof, or patented or described in any printed publication in any country before my invention thereof or more than one year prior to this application, that the same was not in public use or on sale in the United States of America more than one year prior to this application, and that the invention has not been patented or made the subject of an inventor's certificate issued before the date of this application in any country foreign to the United States of America on an application filed by me or my legal representatives or assigns more than twelve months (for a utility patent application) or six months (for a design patent application) prior to this application.

I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d), of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s):

APPLICATION NUMBER	COUNTRY (OR INDICATE IF PCT)	DATE OF FILING (day, **month**, year)	PRIORITY CLAIMED UNDER 37 USC 119
			☐ No ☐ Yes
			☐ No ☐ Yes
			☐ No ☐ Yes

I hereby claim the benefit under Title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below:

APPLICATION NUMBER	FILING DATE

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application:

APPLICATION NUMBER	FILING DATE	STATUS (ISSUED, PENDING, ABANDONED)

I hereby appoint the persons listed on Appendix A hereto (which is incorporated by reference and a part of this document) as my respective patent attorneys and patent agents, with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith.

Send correspondence to:
Thinh V. Nguyen, Reg. No. 42,034, BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP
(Name of Attorney or Agent)
12400 Wilshire Boulevard, 7th Floor, Los Angeles, California 90025 and direct telephone calls to:
Thinh V. Nguyen, (714) 557-3800.
(Name of Attorney or Agent)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Sole/First Inventor (given name, family name) **Carl M. Ellison**

Inventor's Signature _____ Date _____

Residence Portland, Oregon USA Citizenship USA
(City , State) *(Country)*

P. O. Address 1818 N.W. 28th Avenue

Portland, Oregon 97210-2214 USA

Full Name of Second/Joint Inventor (given name, family name) **Roger A. Golliver**

Inventor's Signature _____ Date _____

Residence Beaverton, Oregon USA *(City , State)* Citizenship USA *(Country)*

P. O. Address 13340 S. W. Violet Ct.

Beaverton, Oregon 97008 USA

Full Name of Third/Joint Inventor (given name, family name) **Howard C. Herbert**

Inventor's Signature _____ Date _____

Residence Phoenix, Arizona USA *(City , State)* Citizenship USA *(Country)*

P. O. Address 16817 South 1st Drive

Phoenix, Arizona 85045 USA

Full Name of Fourth/Joint Inventor (given name, family name) **Derrick C. Lin**

Inventor's Signature _____ Date _____

Residence Foster City, California USA *(City , State)* Citizenship USA *(Country)*

P. O. Address 113 Barkentine Street

Foster City, California 94404 USA

Full Name of Fifth/Joint Inventor (given name, family name) **Francis X. McKeen**

Inventor's Signature _____ Date _____

Residence Portland, Oregon USA *(City , State)* Citizenship USA *(Country)*

P. O. Address 10612 N. W. LeMans Ct.

Portland, Oregon 97229 USA

Full Name of Sixth/Joint Inventor (given name, family name) **Gil Neiger**

Inventor's Signature _____ Date _____

Residence Portland, Oregon USA (City, State) Citizenship USA (Country)

P. O. Address 2424 N. E. 11th Avenue

Portland, Oregon 97212 USA

Full Name of Seventh/Joint Inventor (given name, family name) **Ken Reneris**

Inventor's Signature _____ Date _____

Residence Wilbraham, Massachusetts USA (City, State) Citizenship USA (Country)

P. O. Address 8 Red Gap Road

Wilbraham, Massachusetts 01095 USA

Full Name of Eighth/Joint Inventor (given name, family name) **James A. Sutton**

Inventor's Signature _____ Date _____

Residence Portland, Oregon USA (City, State) Citizenship USA (Country)

P. O. Address 20205 N. W. Paulina Drive

Portland, Oregon 97229 USA

Full Name of Ninth/Joint Inventor (given name, family name) **Shreekant S. Thakkar**

Inventor's Signature _____ Date _____

Residence Portland, Oregon USA (City, State) Citizenship United Kingdom (Country)

P. O. Address 150 S. W. Moonridge Place

Portland, Oregon 97225 USA

Full Name of Tenth/Joint Inventor (given name, family name) **Millind Mittal**

Inventor's Signature _____ Date _____

Residence Palo Alto, California USA Citizenship USA
(City , State) *(Country)*

P. O. Address 800 E. Charleston Road #29

Palo Alto, California 94303 USA

Full Name of Eleventh/Joint Inventor (given name, family name) _____

Inventor's Signature _____ Date _____

Residence _____ Citizenship _____
(City , State) *(Country)*

P. O. Address _____

_____

DO NOT PRINT ON THIS LINE

APPENDIX A

I hereby appoint BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP, a firm including: William E. Alford, Reg. No. 37,764; Farzad E. Amini, Reg. No. 42,261; Amy M. Armstrong, Reg. No. 42,265; Aloysius T. C. AuYeung, Reg. No. 35,432; William Thomas Babbitt, Reg. No. 39,591; Carol F. Barry, Reg. No. 41,600; Jordan Michael Becker, Reg. No. 39,602; Bradley J. Bereznak, Reg. No. 33,474; Michael A. Bernadicou, Reg. No. 35,934; Roger W. Blakely, Jr., Reg. No. 25,831; Gregory D. Caldwell, Reg. No. 39,926; Ronald C. Card, Reg. No. 44,587; Thomas M. Coester, Reg. No. 39,637; Donna Jo Coningsby, Reg. No. 41,684; Michael Anthony DeSanctis, Reg. No. 39,957; Daniel M. De Vos, Reg. No. 37,813; Robert Andrew Diehl, Reg. No. 40,992; Matthew C. Fagan, Reg. No. 37,542; Tarek N. Fahmi, Reg. No. 41,402; George L. Fountain, Reg. No. 36,374; Paramita Ghosh, Reg. No. 42,806; James Y. Go, Reg. No. 40,621; James A. Henry, Reg. No. 41,064; Willmore F. Holbrow III, Reg. No. 41,845; Sheryl Sue Holloway, Reg. No. 37,850; George W Hoover II, Reg. No. 32,992; Eric S. Hyman, Reg. No. 30,139; William W. Kidd, Reg. No. 31,772; Sang Hui Kim, Reg. No. 40,450; Eric T. King, Reg. No. 44,188; Erica W. Kuo, Reg. No. 42,775; Michael J. Mallie, Reg. No. 36,591; Paul A. Mendonsa, Reg. No. 42,879; Darren J. Milliken, Reg. No. 42,004; Chun M. Ng, Reg. No. 36878; Thien T. Nguyen, Reg. No. 43,835; Thinh V. Nguyen, Reg. No. 42,034; Dennis A. Nicholls, Reg. No. 42,036; Lisa A. Norris, Reg. No. 44,976; Daniel E. Ovanezian, Reg. No. 41,236; William F. Ryann, Reg. No. 44,313; James H. Salter, Reg. No. 35,668; William W. Schaal, Reg. No. 39,018; James C. Scheller, Reg. No. 31,195; Jeffrey S. Smith, Reg. No. 39,377; Maria McCormack Sobrino, Reg. No. 31,639; Stanley W. Sokoloff, Reg. No. 25,128; Judith A. Szepesi, Reg. No. 39,393; Vincent P. Tassinari, Reg. No. 42,179; Edwin H. Taylor, Reg. No. 25,129; Joseph A. Twarowski, Reg. No. 42,191; Lester J. Vincent, Reg. No. 31,460; Glenn E. Von Tersch, Reg. No. 41,364; John Patrick Ward, Reg. No. 40,216; Charles T. J. Weigell, Reg. No. 43,398; James M. Wu, Reg. No. 45,241; Steven D. Yates, Reg. No. 42,242; and Norman Zafman, Reg. No. 26,250; my attorneys; and Andrew C. Chen, Reg. No. 43,544; Justin M. Dillon, Reg. No. 42,486; and John F. Travis, Reg. No. 43,203; my patent agents, of BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP, with offices located at 12400 Wilshire Boulevard, 7th Floor, Los Angeles, California 90025, telephone (714) 557-3800, and Alan K. Aldous, Reg. No. 31,905; Robert D. Anderson, Reg. No. 33,826; Joseph R. Bond, Reg. No. 36,458; Richard C. Calderwood, Reg. No. 35,468; Jeffrey S. Draeger, Reg. No. 41,000; Cynthia Thomas Faatz, Reg No. 39,973; Sean Fitzgerald, Reg. No. 32,027; John N. Greaves, Reg. No. 40,362; Seth Z. Kalson, Reg. No. 40,670; David J. Kaplan, Reg. No. 41,105; Charles A. Mirho, Reg. No. 41,199; Leo V. Novakoski, Reg. No. 37,198; Naomi Obinata, Reg. No. 39,320; Thomas C. Reynolds, Reg. No. 32,488; Kenneth M. Seddon, Reg. No. 43,105; Mark Seeley, Reg. No. 32,299; Steven P. Skabrat, Reg. No. 36,279; Howard A. Skaist, Reg. No. 36,008; Steven C. Stewart, Reg. No. 33,555; Raymond J. Werner, Reg. No. 34,752; Robert G. Winkle, Reg. No. 37,474; and Charles K. Young, Reg. No. 39,435; my patent attorneys, and Thomas Raleigh Lane, Reg. No. 42,781; Calvin E. Wells; Reg. No. P43,256, Peter Lam, Reg. No. 44,855; and Gene I. Su, Reg. No. 45,140; my patent agents, of INTEL CORPORATION; and James R. Thein, Reg. No. 31,710, my patent attorney; with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith.